

1 What is claimed is:

2 1. A system comprising:

3 a first port to receive a network packet;

4 a second port in communication with the first port, the
5 second port to transmit the network packet after processing;

6 circuitry to associate first control information with a
7 first portion of the network packet and to associate second
8 control information with a second portion of the network
9 packet;

10 circuitry to process the first portion of the network
11 packet and to process the second portion of the network packet
12 at least partially in parallel with processing the first
13 portion of the network packet; and

14 circuitry to enqueue the first portion and the second
15 portion for transmission to a second port in the same order in
16 which the first portion and the second portion were received
17 at the first port.

18

19 2. The system of claim 1 wherein the circuitry
20 comprises:

21 one or more peripheral buses;

22 a memory system;

23 a processor coupled to the one or more peripheral buses
24 and the memory system, the processor adapted to forward data
25 from the first port to the second port; and
26 a bus interface to receive the first portion of the
27 network packet and the second portion of the network packet
28 from the first port and enqueueing the first portion and the
29 second portion in the order in which they were received from
30 the first port for transmission to the second port, the first
31 and second portions being processed at least partially in
32 parallel.

33

34 3. The system of claim 2 wherein the processor
35 comprises one or more microengines to execute program threads,
36 the threads include receive schedule program threads to assign
37 the first portion of the network packet from the first port to
38 a first receive processing program thread and the second
39 portion of the network packet to a second receive processing
40 program thread, wherein the bus interface is responsive to the
41 one or more microengines, and wherein the first and second
42 receive processing program threads are adapted for processing
43 and enqueueing.

44

45 4. The system of claim 3 wherein the bus interface uses
46 sequence numbers to enqueue the first portion and the second

47 portion, wherein the bus interface is to associate one or more
48 first portion sequence numbers with the first portion and one
49 or more second portion sequence numbers with the second
50 portion as the first and second portions are received from the
51 first port.

52

53 5. The system of claim 4 wherein the bus interface is
54 further to maintain a second set of sequence numbers for use
55 by the first and second receive processing program threads in
56 determining the order in which the first and second portions
57 are to be enqueued.

58

59 6. The system of claim 2 wherein the one or more
60 peripheral buses comprise at least one input-output bus,
61 wherein the processor is adapted to interface over the input-
62 output bus with at least one of a media access controller
63 device and a high-speed device, the high-speed device
64 comprising at least one of a gigabit Ethernet MAC and a dual
65 gigabit MAC with two ports.

66

67 7. The system of claim 2 wherein the memory system
68 further comprises at least one of a random access memory, a
69 synchronous dynamic random access memory, a synchronous

dynamic random access memory controller, a static random access memory controller, and a nonvolatile memory.

8. The system of claim 7 wherein the memory system further comprises a memory bus, wherein the memory bus is adapted to couple one or more bus interfaces to one or more memory controllers.

9. The system of claim 2 wherein the processor comprises one or more microengines to execute program threads, wherein the one or more microengines are configured to operate with shared resources, and wherein the shared resources comprise the memory system and the one or more peripheral buses.

10. The system of claim 9 wherein the bus interface comprises an input-output bus interface.

11. The system of claim 9 wherein the bus interface is coupled to an input-output bus, wherein the input-output bus is coupled to a dual gigabit MAC.

12. The system of claim 9 wherein at least one of the microengines comprises:

94 a control store for storing a microprogram; and
95 a set of control logic, wherein the set of control logic
96 comprises an instruction decoder and one or more program
97 counter units.

98

99 13. The system of claim 12 wherein at least one of the
100 microengines further comprises a set of context event
101 switching logic to receive messages from the shared resources.

102

103 14. A communication system comprising:

104 a media access controller capable of providing one or
105 more status flags, the media access controller comprising one
106 or more ports;

107 a bus interface unit comprising one or more registers,
108 wherein the one or more registers comprise control registers
109 and status registers;

110 a bus connected between the media access controller and
111 the bus interface unit; and

112 a sequencer to poll the one or more status flags and
113 place the one or more status flags to the one or more
114 registers over the bus, wherein the communication system is
115 capable of processing one or more packets of data, and wherein
116 the communication system is capable of maintaining an intra-

117 packet order and an inter-packet order for the one or more
118 ports.

119

120 15. The communication system of claim 13 wherein the
121 media access controller further comprises one or more transmit
122 registers and one or more receive registers, and wherein the
123 one or more ports comprise at least two gigabit Ethernet
124 ports.

125

126 16. The communication system of claim 15 wherein the
127 communication system is capable of enqueueing a first portion
128 of a network packet and a second portion of a network packet
129 for transmission to a second port in the same order in which
130 the first portion and the second portion were received at a
131 first port.

132

133 17. The communication system of claim 16 wherein the one
134 or more status flags comprise one or more transmit status
135 flags and one or more receive status flags, and wherein the
136 one or more flags indicate whether an amount of data in
137 associated transmit registers and associated received
138 registers have reached a threshold level.

139

140 18. The communication system of claim 17 wherein a
141 receive scheduler thread uses the one or more registers in the
142 bus interface unit to determine how to issue a receive
143 request.

144

145 19. The communication system of claim 16 wherein the
146 communication system uses a set of sequence numbers for each
147 port, wherein the sequence numbers comprise a network packet
148 sequence number, a MAC packet sequence number, and an enqueue
149 sequence number.

150

151 20. A method of supporting one or more network port
152 modes comprising:

153 determining an assignment of one or more receive
154 processing threads to process packet data in a queue, wherein
155 the one or more network port modes comprise a single thread
156 mode, a header/body thread mode, and an explicit thread mode;

157 determining an availability of one or more threads to
158 process a received request;

159 determining an execution time for the one or more
160 received threads; and

161 determining an end of a network packet and a beginning of
162 a successive network packet.

163

164 21. The method of claim 20 wherein the single thread
165 mode assigns a single thread to a packet for one or more
166 speculative requests.

167

168 22. The method of claim 20 wherein the header/body mode
169 assigns two threads to process a MKPT within the network
170 packet, wherein the first thread comprises the header thread
171 and the second thread is the body thread, and wherein the
172 header thread is used to determine how to forward the network
173 packet and the body thread is used for moving a remainder of
174 the network packet to memory.

175

176 23. The method of claim 20 wherein the explicit thread
177 mode comprises providing one or more received processing
178 threads with an identification of a thread assigned to the
179 next MPKT receive request, wherein each assigned thread
180 provides the next assigned thread with a pointer to a memory.

181

182 24. A system comprising:
183 receiving means to receive a network packet at a first
184 port;

185 transmitting means for transmitting the network packet
186 after processing, the receiving means in communication with
187 the transmitting means;

188 means for associating a first control information with a
189 first portion of the network packet;
190 means for associating second control information with a
191 second portion of the network packet;
192 means for processing the first portion of the network
193 packet and the second portion of the network packet at least
194 partially in parallel; and
195 means for enqueueing the first portion and the second
196 portion for transmitting to a second port in the same order in
197 which the first portion and the second portion were received
198 at the first port.

199

200 25. The system of claim 24 wherein the means for
201 processing the first portion of the network packet and the
202 second portion of the network packet at least partially in
203 parallel is implemented at least partially in software.